CONCEPTS RELATED TO DOM MANIPULATION

DOM Selection

- document.getElementById()
- document.getElementsByClassName()
- document.getElementsByTagName()
- document.querySelector()
- document.querySelectorAll()
- parentNode.querySelector()
- parentNode.querySelectorAll()

HTML

```html
<div id="myDiv">
  <p>Hello, World!</p>
  <p>DOM Selection Example</p>
</div>
```

javascript

```javascript
const myDiv = document.getElementById("myDiv");
constparagraphs = document.getElementsByTagName("p");
const firstParagraph = document.querySelector("p");
const allParagraphs = document.querySelectorAll("p");
```

DOM Traversal

- parentNode.childNodes
- parentNode.firstChild
- parentNode.lastChild
- element.nextSibling
- element.previousSibling
- parentNode.children
- element.parentElement
- element.closest()

HTML

```html
<ul id="list">
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
</ul>
```

javascript

```javascript
const list = document.getElementById("list");
const firstListItem = list.firstChild;
const lastListItem = list.lastChild;
const secondListItem = firstListItem.nextSibling;
const parentOfList = list.parentElement;
const closestDiv = list.closest("div");
```

DOM Modification

- element.textContent
- element.innerHTML
- element.setAttribute()
- element.removeAttribute()
- element.classList.add()
- element.classList.remove()
- element.classList.toggle()
- element.style

HTML

```html
<p id="myParagraph">Hello, DOM Manipulation!</p>
```

javascript

```javascript
const myParagraph =
document.getElementById("myParagraph");
myParagraph.textContent = "Hello, World!";
myParagraph.innerHTML = "<strong>Hello, World!
</strong>"; myParagraph.setAttribute("class",
"highlight"); myParagraph.removeAttribute("id");
myParagraph.classList.add("new-class");
myParagraph.classList.remove("old-class");
myParagraph.classList.toggle("highlight");
myParagraph.style.color = "blue";
```

Creating and Appending Elements

- document.createElement()
- document.createTextNode()
- parentNode.appendChild()
- parentNode.insertBefore()
- parentNode.replaceChild()

HTML

```html
<ul id="todoList">
<li>Item 1</li>
<li>Item 2</li>
</ul>
```

javascript

```javascript
const newTodoItem = document.createElement("li");
newTodoItem.textContent = "Item 3";
const todoList = document.getElementById("todoList");
todoList.appendChild(newTodoItem);
```

Removing Elements

- element.remove()
- parentNode.removeChild()

HTML

```html
<div id="container">
<p>Content to be removed</p>
</div>
```

javascript

```javascript
const container = document.getElementById("container");
const paragraphToRemove = container.querySelector("p");
paragraphToRemove.remove();
```

Event Handling

- element.addEventListener()
- element.removeEventListener()
- event.preventDefault()
- event.stopPropagation()
- event.currentTarget
- event.target

HTML

```html
<button id="myButton">Click Me</button>
```

javascript

```javascript
const myButton = document.getElementById("myButton");
myButton.addEventListener("click", function() {
 alert("Button clicked!");
 });
```

CSS Classes and Styling

- element.className
- element.classList
- element.style

HTML

```html
<p id="myPara" class="highlight">Hello, CSS Classes!</p>
```

javascript

```javascript
const myPara = document.getElementById("myPara");
myPara.className = "new-class";
myPara.classList.add("highlight");
myPara.classList.remove("highlight");
myPara.classList.toggle("highlight");
myPara.style.color = "blue";
```

DOM Manipulation and Asynchronous Operations

- Handling AJAX requests and updating the DOM with fetched data.

```javascript
fetch("https://api.example.com/data")
.then(response => response.json())
.then(data => {
const myDiv = document.getElementById("myDiv");
myDiv.textContent = data.message; })
.catch(error => console.error(error));
```

Was this helpful to you?

Please like and share it